

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 2  
CLASS & OBJECT**



Oleh:

Ramadhan Wijaya (2211102208)

Dosen Pengampu :

Dedy Agung Prabowo, S.Kom., M.Kom.

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

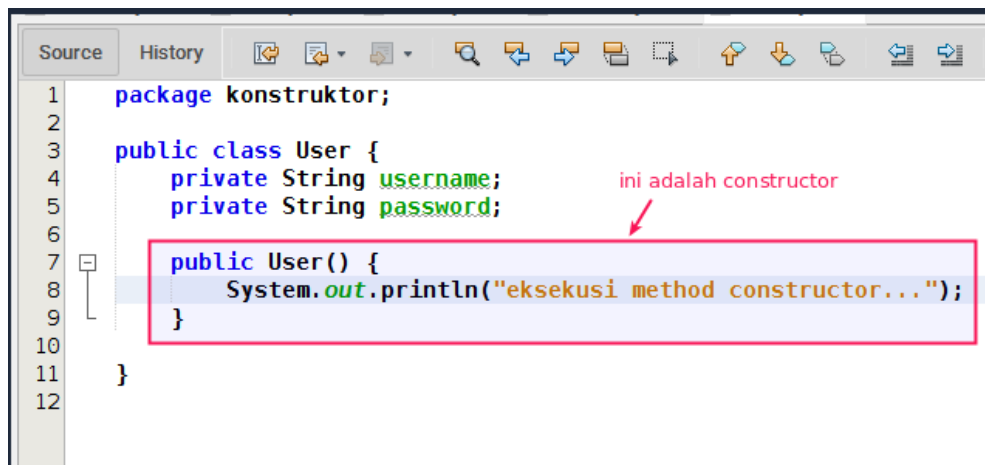
**2024**

## I. DASAR TEORI

Class adalah kumpulan dari variable dan fungsi/procedure yang dikumpulkan menjadi satu dengan tujuan membuat sebuah object, bisa dibilang blueprint dari sebuah object. Sedangkan object adalah sebuah variable yang instance perwujudan dari class. dalam Class variable disebut atribut/Fields, dan fungsi disebut method.

Fields atau variable adalah implementasi dari atribut suatu objek. Field atau variable digunakan untuk menyimpan objek

konstruktor adalah method khusus yang akan dieksekusi saat objek dibuat, biasanya method ini digunakan untuk inisialisasi atau mempersiapkan data untuk objek. contoh konstruktor adalah sebagai berikut :



```
1 package konstruktor;
2
3 public class User {
4     private String username;
5     private String password;
6
7     public User() {
8         System.out.println("eksekusi method constructor...");
9     }
10
11 }
12
```

ketika membuat konstruktor pastikan methodnya public kepada konstruktor, karena ia akan dieksekusi saat membuat objek.

selain constructor ada juga destructor tetapi karena java menggunakan sistem garbage collector jadi bisa menghapus objek yang tidak terpakai, penggunaan destructor ini berada di bahasa pemrograman c++. contoh destructor seperti dibawah ini :

```

class User {
public:
    User( String *username ); // <-- ini constructor
    ~User(); // <-- ini destructor.
private:
    String username;
    String password;
};

```

method atau bisa disebut fungsi/prosedur memiliki tujuan memecah program kompleks menjadi bagian bagian kecil sehingga nantinya dapat kita gunakan berulang ulang tanpa harus menulis baris kode yang sama. contoh program method :

```

public void cetakPesan(){
    System.out.println("Belajar Java bersama kodedasar.com");
}

```

## II. GUIDED

### BUKU.CPP

```
//RAMADHAN WIJAYA 2211102208
#include <iostream>
#include <cstring>

using namespace std;

class Buku {
private:
    char judul[35];
    char pengarang[25];
    int jumlah;

public:
    void inisialisasiBuku(const char *jdl, const char
    *pngarang, int jmlh) {
        strcpy(judul, jdl);
        strcpy(pengarang, pngarang);
        jumlah = jmlh;
    }

    void infoBuku() {
        cout << " Judul :" << judul << endl;
        cout << " Pengarang :" << pengarang << endl;
        cout << " Jumlah buku :" << jumlah << endl;
    }
};

int main() {
    Buku novel, fiksi;

    novel.inisialisasiBuku("Meriam Benteng navarone",
    "Alistair Maclean", 12);
    fiksi.inisialisasiBuku("Jurassic park", "Michael
    Crichton", 3);

    novel.infoBuku();
    fiksi.infoBuku();

    getchar(); // Menunggu input sebelum keluar
    return 0;
}
```

## OUTPUT:

```
PowerShell 7.4.1
PS D:\SEMESTER_3\Praktikum_PBO> & 'c
:\Users\ramad\.vscode\extensions\ms-v
scode.cpptools-1.19.6-win32-x64\debug
Adapters\bin\WindowsDebugLauncher.exe
' '--stdin=Microsoft-MIEngine-In-vqhi
i2p2.nlz' '--stdout=Microsoft-MIEngin
e-Out-kjr5xpgv.3jh' '--stderr=Microso
ft-MIEngine-Error-yvawlzik.1xq' '--pi
d=Microsoft-MIEngine-Pid-pd0mhdwv.vpl
' '--dbgExe=C:\MinGW\bin\gdb.exe' '--
interpreter=mi'
Judul :Meriam Benteng navarone
Pengarang :Alistair Maclean
Jumlah buku :12
Judul :Jurassic park
Pengarang :Michael Crichton
Jumlah buku :3
```

## KETERANGAN :

Program C++ di atas adalah implementasi sederhana dari suatu kelas Buku yang digunakan untuk merepresentasikan informasi buku. Program ini menggunakan objek dari kelas Buku untuk menginisialisasi dan menampilkan informasi dua buku.

## BUKU.JAVA

```
package BUKU.JAVA;

/**
 *
 * @author 2211102208
 */
class Buku {

    private String judul;
    private String pengarang;
    private int jumlah;

    public void setNilai(String judul, String pengarang, int jumlah) {
        this.judul = judul;
        this.pengarang = pengarang;
        this.jumlah = jumlah;
    }
}
```

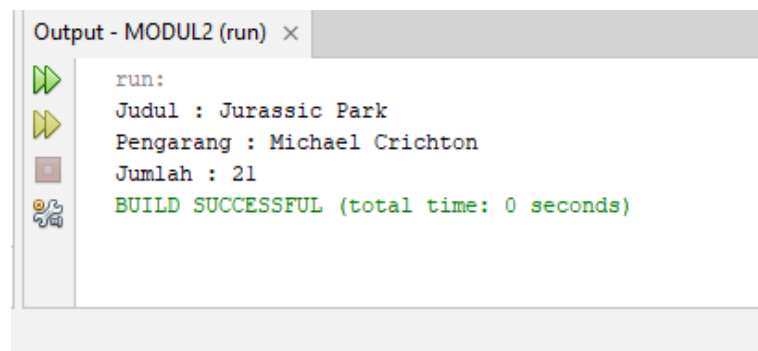
```

void cetakKeLayar() {
    System.out.println("Judul : " + judul);
    System.out.println("Pengarang : " + pengarang);
    System.out.println("Jumlah : " + jumlah);
}

public static void main(String[] args) {
    Buku a = new Buku();
    a.setNilai("Jurassic Park", "Michael Crichton", 21);
    a.cetakKeLayar();
}
}

```

OUTPUT :



```

Output - MODUL2 (run) x
run:
Judul : Jurassic Park
Pengarang : Michael Crichton
Jumlah : 21
BUILD SUCCESSFUL (total time: 0 seconds)

```

KETERANGAN :

Sama dengan program BUKU.cpp diatas mengimplementasikan program buku menggunakan objek dan kelas bedanya hanya dalam bahasa, disini kita menggunakan bahasa pemrograman JAVA.

KOMPLEKS.CPP

```

//RAMADHAN WIJAYA 2211102208
#include <iostream>
class Kompleks
{
private:
    double re, im;

public:
    // Konstruktor default

```

```

    Kompleks()
    {
        std::cout << "Konstruktor Kompleks dijalankan...."
<< std::endl;
        re = 5.2;
        im = 3.6;
    }

    // Konstruktor dengan parameter
    Kompleks(double real, double imajiner)
    {
        std::cout << "Konstruktor Kompleks dijalankan...."
<< std::endl;
        re = real;
        im = imajiner;
    }

    // Metode info untuk menampilkan informasi bilangan
    kompleks
    void info()
    {
        std::cout << "\nBilangan kompleks : " << std::endl;
        std::cout << "real = " << re << std::endl;
        std::cout << "imajiner = " << im << std::endl;
        std::cout << std::endl;
    }
};

int main()
{
    Kompleks a;           // Memanggil konstruktor default
    a.info();             // Menampilkan informasi bilangan
    kompleks a
    Kompleks b(3.2, 4.1); // Memanggil konstruktor dengan
    parameter
    b.info();             // Menampilkan informasi bilangan
    kompleks b

    return 0;
}

```

OUTPUT :

```

Konstruktor Kompleks dijalankan....

Bilangan kompleks :
real = 5.2
imajiner = 3.6

Konstruktor Kompleks dijalankan....

Bilangan kompleks :
real = 3.2
imajiner = 4.1

PS D:\SEMESTER_3\Praktikum_PBO>

```

#### KETERANGAN :

Jadi pada program diatas adalah program menampilkan bilangan bilangan dengan 2 konstruktor Bernama kompleks dan void info untuk menampilkan semua konstruktor. Pada int main void info menampilkan data dari deklarasi sekaligus inisialisasi yang original, Ketika void info yang kedua menampilkan data yang sudah user masukan sendiri secara langsung di kode program.

#### BUKU2.JAVA

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
 * default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
 * to edit this template
 */

/**
 *
 * @author 2211102208
 */
package ProjectBuku2;

public class BUKU {

    static class Buku {

        private String judul;
        private String pengarang;
    }
}

```



```

public Buku() {
    judul = "Tidak diketahui";
    pengarang = "Tidak diketahui";
}

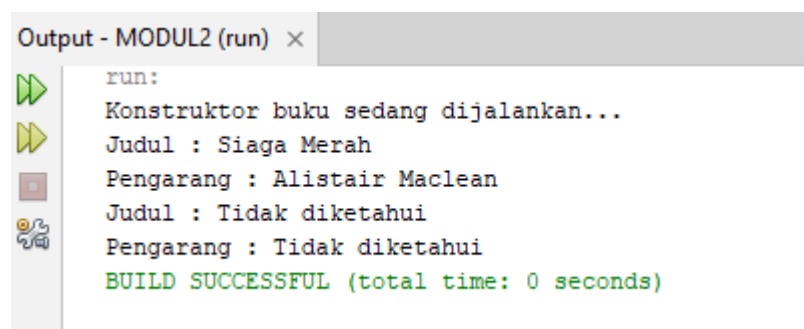
public Buku(String judul, String pengarang) {
    System.out.println("Konstruktor buku sedang dijalankan...");
    this.judul = judul;
    this.pengarang = pengarang;
}

void cetakKeLayar() {
    if (judul == null && pengarang == null) {
        return;
    }
    System.out.println("Judul : " + judul);
    System.out.println("Pengarang : " + pengarang);
}

public static void main(String[] args) {
    Buku a, b;
    a = new Buku("Siaga Merah", "Alistair Maclean");
    b = new Buku();
    a.cetakKeLayar();
    b.cetakKeLayar();
}
}

```

## OUTPUT :



```

Output - MODUL2 (run) x
run:
Konstruktor buku sedang dijalankan...
Judul : Siaga Merah
Pengarang : Alistair Maclean
Judul : Tidak diketahui
Pengarang : Tidak diketahui
BUILD SUCCESSFUL (total time: 0 seconds)

```

## KETERANGAN :

Program diatas membuat sebuah kelas buku dan method yang Namanya juga buku. Ada method yang harus di inputkan secara manual dan ada yang sudah terisi dengan “tidak diketahui”. Pada bagian main program variable a dimasukan

method yang yang harus diisikan secara manual sedangkan yang B langsung mengeluarkan “tidak diketahui”

### III. UNGUIDED

#### Unguided 1

##### **BANK.java**

```
/**
 *
 * @author 2211102208
 */
class BANK {
    private int saldo;

    // Konstruktor class Bank dengan parameter saldo
    public BANK(int saldo) {
        this.saldo = saldo;
    }

    // Method untuk menyimpan uang ke saldo
    public void simpanUang(int jumlah) {
        saldo += jumlah;
    }

    // Method untuk mengambil uang dari saldo
    public void ambilUang(int jumlah) {
        if (jumlah <= saldo) {
            saldo -= jumlah;
        } else {
            System.out.println("Maaf, saldo tidak mencukupi.");
        }
    }

    // Method untuk mendapatkan saldo
    public int getSaldo() {
        return saldo;
    }
}
```

##### **BankBeraksi.java**

```
package UNGUIDED;
```

```
/**
```

```

*
* @author 2211102208
*/
public class BankBeraksi {
    public static void main(String[] args) {
        // Membuat objek Bank dengan saldo awal Rp. 100000
        BANK bankABC = new BANK(100000);

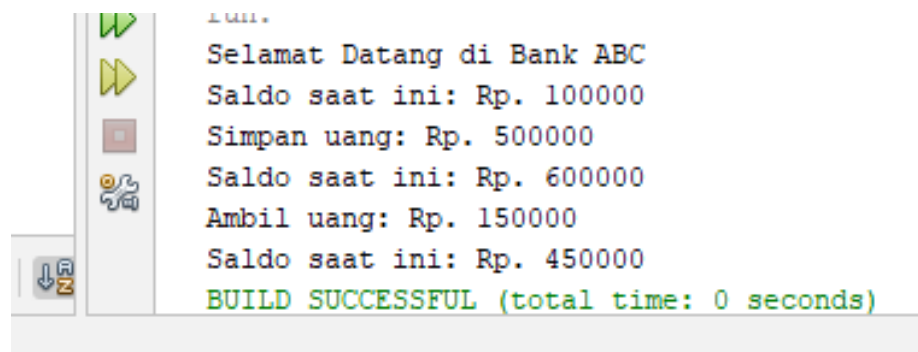
        // Menampilkan pesan selamat datang dan saldo awal
        System.out.println("Selamat Datang di Bank ABC");
        System.out.println("Saldo saat ini: Rp. " +
bankABC.getSaldo());

        // Menyimpan uang Rp. 500000
        bankABC.simpanUang(500000);
        System.out.println("Simpan uang: Rp. 500000");
        System.out.println("Saldo saat ini: Rp. " +
bankABC.getSaldo());

        // Mengambil uang Rp. 150000
        bankABC.ambilUang(150000);
        System.out.println("Ambil uang: Rp. 150000");
        System.out.println("Saldo saat ini: Rp. " +
bankABC.getSaldo());
    }
}

```

Output :



```

1. Main
Selamat Datang di Bank ABC
Saldo saat ini: Rp. 100000
Simpan uang: Rp. 500000
Saldo saat ini: Rp. 600000
Ambil uang: Rp. 150000
Saldo saat ini: Rp. 450000
BUILD SUCCESSFUL (total time: 0 seconds)

```

Keterangan ;

Pada program diatas membuat program sesuai ketentuan di modul dan pada saat ini saya belum paham tentang perkondisian di JAVA, jadi saya membuat secara manual tanpa perkondisian.

## Unguided 2

```
Graph.java
/*
 * Click
 nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
 default.txt to change this license
 * Click
 nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.jav
 a to edit this template
 */
package UNGUIDED2;

/**
 *
 * @author 2211102208
 */
import java.util.Scanner;

class GRAPH {

    private int x;
    private int y;

    public GRAPH() {
        this.x = 0;
        this.y = 0;
    }

    public void inisialisasiTitik(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public void tampilTitik() {
```

```

        System.out.println("Titik: (" + x + ", " + y + ")");
    }

    public void perkalianSkalar(int skalar) {
        this.x *= skalar;
        this.y *= skalar;
    }

    public void pencerminanSumbuX() {
        this.y = -this.y;
    }

    public void pencerminanSumbuY() {
        this.x = -this.x;
    }

    public double jarak(GRAPH t2) {
        int dx = t2.x - this.x;
        int dy = t2.y - this.y;
        return Math.sqrt(dx * dx + dy * dy);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        GRAPH titik1 = new GRAPH();
        GRAPH titik2 = new GRAPH();

        int choice;
        do {
            System.out.println("\nMENU OPERASI TITIK");
            System.out.println("1. INISIALISASI TITIK");
            System.out.println("2. TAMPIL TITIK");
            System.out.println("3. PERKALIAN SKALAR
TITIK");
            System.out.println("4. PENCERMINAN TERHADAP
SUMBU X");
            System.out.println("5. PENCERMINAN TERHADAP
SUMBU Y");

```

```

        System.out.println("6. JARAK ANTARA DUA
TITIK");
        System.out.println("0. KELUAR");
        System.out.print("MASUKAN PILIHAN : ");
        choice = scanner.nextInt();

        switch (choice) {
            case 1:
                System.out.print("\nMasukkan koordinat x titik
1: ");

                int x1 = scanner.nextInt();
                System.out.print("Masukkan koordinat y titik 1:
");

                int y1 = scanner.nextInt();
                titik1.inisialisasiTitik(x1, y1);

                System.out.print("\nMasukkan koordinat x titik
2: ");

                int x2 = scanner.nextInt();
                System.out.print("Masukkan koordinat y titik 2:
");

                int y2 = scanner.nextInt();
                titik2.inisialisasiTitik(x2, y2);
                break;
            case 2:
                System.out.println("\nTitik 1:");
                titik1.tampilTitik();
                System.out.println("Titik 2:");
                titik2.tampilTitik();
                break;
            case 3:
                System.out.print("\nMasukkan nilai skalar: ");
                int skalar = scanner.nextInt();
                titik1.perkalianSkalar(skalar);
                titik2.perkalianSkalar(skalar);
                System.out.println("\nTitik 1 setelah perkalian
skalar:");
                titik1.tampilTitik();

```

```

        System.out.println("Titik 2 setelah perkalian
skalar:");
        titik2.tampilTitik();
        break;
    case 4:
        titik1.pencerminanSumbuX();
        titik2.pencerminanSumbuX();
        System.out.println("\nTitik 1 setelah
pencerminan sumbu X:");
        titik1.tampilTitik();
        System.out.println("Titik 2 setelah pencerminan
sumbu X:");
        titik2.tampilTitik();
        break;
    case 5:
        titik1.pencerminanSumbuY();
        titik2.pencerminanSumbuY();
        System.out.println("\nTitik 1 setelah
pencerminan sumbu Y:");
        titik1.tampilTitik();
        System.out.println("Titik 2 setelah pencerminan
sumbu Y:");
        titik2.tampilTitik();
        break;
    case 6:
        System.out.println("\nJarak antara titik 1 dan titik
2: " + titik1.jarak(titik2));
        break;
    case 0:
        System.out.println("\nTerima kasih!");
        break;
    default:
        System.out.println("\nPilihan tidak valid!");
    }
} while (choice != 0);

scanner.close();
}
}

```



## Output :

The image displays two screenshots of the Apache NetBeans IDE, showing the output of a Java program. The program appears to be a geometry application, likely calculating the distance between two points (Titik 1 and Titik 2) after a scalar multiplication.

**First Screenshot (Top):**

- Output - MODUL2 (run):**

```
run:
MENU OPERASI TITIK
1. INISIALISASI TITIK
2. TAMPIL TITIK
3. PERKALIAN SKALAR TITIK
4. PENCERMINAN TERHADAP SUMBU X
5. PENCERMINAN TERHADAP SUMBU Y
6. JARAK ANTARA DUA TITIK
0. KELUAR
MASUKAN PILIHAN : 1

Masukkan koordinat x titik 1: 3
Masukkan koordinat y titik 1: 8

Masukkan koordinat x titik 2: 2
Masukkan koordinat y titik 2: 4

MENU OPERASI TITIK
1. INISIALISASI TITIK
2. TAMPIL TITIK
3. PERKALIAN SKALAR TITIK
4. PENCERMINAN TERHADAP SUMBU X
5. PENCERMINAN TERHADAP SUMBU Y
6. JARAK ANTARA DUA TITIK
0. KELUAR
MASUKAN PILIHAN : 3

Masukkan nilai skalar: 23

Titik 1 setelah perkalian skalar:
```

**Second Screenshot (Bottom):**

- Output - MODUL2 (run):**

```
run:
MENU OPERASI TITIK
1. INISIALISASI TITIK
2. TAMPIL TITIK
3. PERKALIAN SKALAR TITIK
4. PENCERMINAN TERHADAP SUMBU X
5. PENCERMINAN TERHADAP SUMBU Y
6. JARAK ANTARA DUA TITIK
0. KELUAR
MASUKAN PILIHAN : 3

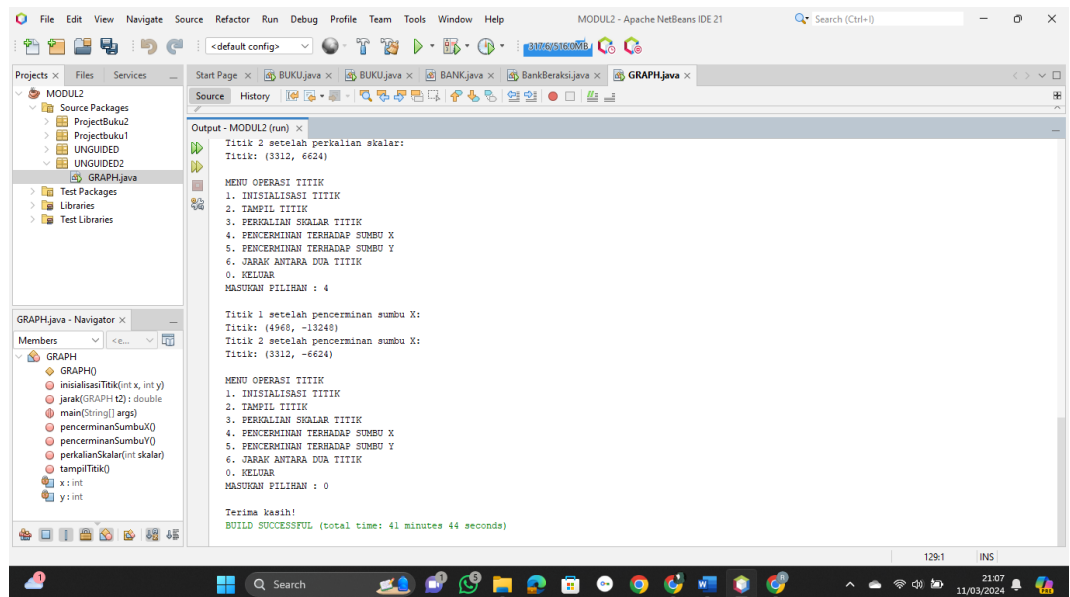
Masukkan nilai skalar: 08

Titik 1 setelah perkalian skalar:
Titik: (552, 1472)
Titik 2 setelah perkalian skalar:
Titik: (368, 736)

MENU OPERASI TITIK
1. INISIALISASI TITIK
2. TAMPIL TITIK
3. PERKALIAN SKALAR TITIK
4. PENCERMINAN TERHADAP SUMBU X
5. PENCERMINAN TERHADAP SUMBU Y
6. JARAK ANTARA DUA TITIK
0. KELUAR
MASUKAN PILIHAN : 3

Masukkan nilai skalar: 9

Titik 1 setelah perkalian skalar:
```



Keterangan :

Kode program Java di atas merupakan implementasi sederhana dari konsep Pemrograman Berorientasi Objek (PBO) yang menggambarkan operasi-operasi pada suatu Titik dalam ruang koordinat. Berikut adalah penjelasan mengenai setiap bagian dari kode program tersebut:

Kelas GRAPH:

Atribut x dan y sebagai koordinat Titik.

1. Default constructor (public GRAPH()) untuk menginisialisasi koordinat Titik menjadi (0,0).
2. Method inisialisasiTitik(int x, int y): Digunakan untuk mengeset nilai koordinat Titik sesuai masukan dari pengguna.
3. Method tampilTitik(): Menampilkan nilai koordinat Titik.
4. Method perkalianSkalar(int skalar): Mengalikan nilai koordinat Titik dengan suatu skalar sesuai masukan dari pengguna.
5. Method pencerminanSumbuX(): Melakukan pencerminan Titik terhadap sumbu X.

6. Method pencerminanSumbuY(): Melakukan pencerminan Titik terhadap sumbu Y.
7. Method jarak(GRAPH t2): Menghitung jarak antara dua Titik.

Metode Utama (main):

1. Membuat objek GRAPH (titik1 dan titik2).
2. Menerapkan sebuah loop do-while untuk menampilkan menu operasi dan melakukan operasi sesuai pilihan pengguna.
3. Menu operasi yang disediakan:
4. Inisialisasi Titik 1 dan Titik 2.
5. Menampilkan koordinat Titik 1 dan Titik 2.
6. Perkalian skalar pada Titik 1 dan Titik 2.
7. Pencerminan terhadap sumbu X pada Titik 1 dan Titik 2.
8. Pencerminan terhadap sumbu Y pada Titik 1 dan Titik 2.
9. Menghitung jarak antara Titik 1 dan Titik 2.
10. Keluar dari program.
11. Pengguna diminta untuk memasukkan pilihan operasi dan data yang diperlukan.

Penutup

Scanner di tutup (scanner.close()) untuk mencegah memory leak.

#### **IV. KESIMPULAN**

- Class merupakan blueprint atau cetakan untuk membuat objek dalam pemrograman Java.
- Class menyediakan struktur, perilaku, dan atribut yang akan dimiliki oleh objek yang akan dibuat.
- Class dapat memiliki atribut (variabel) dan metode (fungsi) yang mendefinisikan karakteristik dan perilaku dari objek yang akan dihasilkan dari class tersebut.
- Atribut menyimpan data, sementara metode melakukan tindakan atau operasi terhadap data tersebut.
- Objek dapat dibuat dengan menginstansiasi (membuat salinan) dari suatu class menggunakan kata kunci new.
- Setelah objek dibuat, atribut dan metode class dapat diakses melalui objek tersebut.

## V. REFRENSI

Belajar Java OOP: Memahami Konsep Dasar OOP (untuk Pemula). (2017). Dirujuk dari <https://www.petanikode.com/javaoop/#:~:text=Class%20adalah%20rancangan%20atau%20blue,fungsi%20yang%20menggambarkan%20sebuah%20objek.>

Belajar Java OOP: Mengenal Constructor & Destructor dalam Java. (2017). Retrieved from <https://www.petanikode.com/java-oop-constructor/>

Pengertian Method: 2+ Contoh Program Fungsi dan Prosedur pada Java. (n.d.). Retrieved from <https://kodedasar.com/blog/method-java/>